Approaches for Language Identification in Mismatched Environments

Shahan Nercessian, Pedro Torres-Carrasquillo, and Gabriel Martínez-Montes

Massachusetts Institute of Technology Lincoln Laboratory

{shahan.nercessian, ptorres, gabriel.martinezmontes}@ll.mit.edu

Abstract

In this paper, we consider the task of language identification in the context of mismatch conditions. Specifically, we address the issue of using unlabeled data in the domain of interest to improve the performance of a state-of-the-art system. The evaluation is performed on a 9-language set that includes data in both conversational telephone speech and narrowband broadcast speech. Multiple experiments are conducted to assess the performance of the system in this condition and a number of alternatives to ameliorate the drop in performance. The best system evaluated is based on deep neural network bottleneck features using i-vectors. The proposed system results in a 30% improvement over the baseline result.

Index Terms: language identification, domain adaptation, unsupervised learning, deep neural networks, bottleneck features

1. Introduction and task

Spoken language identification (LID) is the process of identifying the language in a spoken speech utterance. In recent years, great improvements in LID system performance have been seen due to the advent of new techniques based on low-dimensional feature representations (i-vectors) and more recently, deep neural networks (DNN) and bottleneck features.

Although the observed performance is nothing short of remarkable, issues related to robustness are still of concern. Particularly, the application of LID systems trained on a given set of conditions (domain) but evaluated on a different set of conditions is of interest and results in degraded performance. Conventionally, this problem is addressed by obtaining labeled data in the new domain and retraining the system which usually results in a performance improvement. There are multiple issues with this approach. First, the data available in the new domain may not always be labeled, and labeling the data is a long, manual process that can produce inconsistent labels. Secondly, retraining the system requires interruptions from system operation and potential housekeeping difficulties.

As such, the aim of this paper is two-fold. The primary goal of this work is to explore ways in which unlabeled data can be used in LID systems, particularly with mismatched domains. A secondary goal is to demonstrate the general performance improvement of the hybrid DNN/i-vector system relative to a standard i-vector system.

Distribution A: Public Release; unlimited distribution. This work was sponsored by the Department of Defense under Air Force contract F19628-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

The organization of this paper is as follows: Section 2 describes the experimental setup used to simulate and test the mismatch scenario. Section 3 describes the proposed approaches for addressing the domain mismatch problem. Section 4 presents system performance using the proposed methods, and Section 5 draws conclusions.

2. Experimental setup

2.1. Corpus

The data used for the experiments is a set of 9 languages identified from pooling together most data sources available to us from previous evaluation campaigns and includes only the languages for which data is available in two sources of data: conversational telephone speech (CTS) and broadcast news (BN). The CTS data includes multiple collections from mainly the Linguistic Data Consortium (LDC) while the BN news consists almost exclusively of data collected from Voice of America (VoA). In order to simulate the mismatch scenario, all languages that are available in both CTS and VoA are included in this new subset and used for the mismatch scenario. In particular, the 9 languages that include data from both sources include: Cantonese, Farsi, Hindi, Korean, Mandarin, Russian, Spanish, Urdu and Vietnamese. The experimental setup is to use the CTS data as the available labeled set from which a baseline out-of-domain system is trained. A portion of the VoA data is then used to evaluate in-domain performance with another VoA partition available as unlabeled data. Hereafter, we will use VoA and CTS interchangeably with in- and out-of-domain, respectively.

2.2. LID systems

The mismatched domain alternatives proposed here are assessed using LID systems based on the i-vector framework [1]. Figure 1 illustrates a generic block diagram of the i-vector system used in this paper. Speech is processed following the process described in [2]. For the systems under evaluation, cosine distance scoring is used as the evaluation metric.

In this work, two different i-vector systems are considered, which differ in their feature extraction mechanism. The first, which we refer to as the standard i-vector, or simply i-vector system, uses Shifted Delta Cepstra (SDC) features [3]. The second system uses DNN bottleneck features, and is referred to here as the hybrid DNN/i-vector, or simply DNN system. The bottleneck features are generated by training a DNN to predict senone classification labels generated by Kaldi from frame-level perceptual linear prediction coefficients as input. An intermediate bottleneck layer within the DNN is used as a dimensionality reduction technique, whose outputs serve as frame-level feature vectors. For more info on the DNN system, we refer the reader to [4].

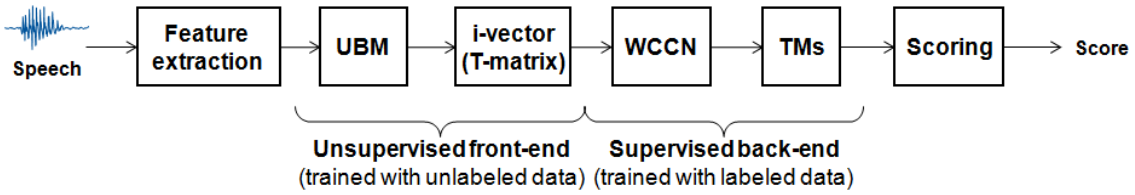

Figure 1: *I-vector LID system block diagram*

3. Proposed approaches

Addressing the mismatched domain problem in this context can be broken into two tasks. The first task involves labeling the unlabeled, but in-domain, training data partition, whereas the second task considers the means by which the newly labeled training data is combined with labeled out-of-domain training data to thereby retrain a new system.

3.1. In-domain data labeling

The first task required in order to address the domain mismatch problem involves accurately labeling the unlabeled VoA partition. As such, it can essentially be viewed as an unsupervised learning problem, in which one expects LID performance to improve with labeling accuracy, and approach best-case system performance with perfect knowledge of the VoA training data labels. The approaches considered here to automatically label the unlabeled VoA training data partition involve the use of an LID system itself. To this effect, three strategies are considered.

1) *Score thresholding*: this approach begins by labeling the VoA data using an entirely CTS-trained LID system. Next, the speech cuts with scores in some top percentile are selected in an attempt to minimize the labeling error. A new LID system is then trained with the retained scores and speech cuts. The notion of the score thresholding is that cuts with higher scores have a higher degree of certainty, as the average class purity increases as the score threshold is increased. Here, we consider the top 50%, 75% and 100% (i.e. all the data) as score thresholds to demonstrate the performance gains of pruning labels in this manner.
2) *Iterative relabeling*: this approach again begins by labeling the VoA data using an entirely CTS-trained LID system. Next, the newly labeled VoA data is incorporated into training a new LID system, which relabels the VoA data, and this process is iterated. The notion is that hopefully, the labeling should be improving with each iteration, improving overall system performance. Relative to the score thresholding approach, the potential advantage of the iterative relabeling approach is that class purity can be improved at each iteration without having to reduce the number of training samples. However, its success is ultimately dependent on the way in which labeling errors propagate over time, and may intrinsically assume that the initial labeling is already quite good.
3) *Hybrid labeling*: this approach is actually a combination of the previous two approaches. Now, the VoA training data is iteratively relabeled, and only the VoA speech cuts with scores above a threshold are retained to retrain a new system.

Note that standard clustering techniques, like k-means, could have alternatively been used to label the data, though we have experimentally found that it is not as fruitful. In some sense, the use of the LID system itself for labeling is simply a more informed unsupervised learning mechanism, though the degree to which this is true will vary with the extent of the mismatch between domains.

3.2. Data pooling

The second task for addressing the mismatch problem involves effectively integrating the newly labeled VoA data to train a new system. To this end, there are three groups of training data that will be used: CTS only, VoA only, and CTS+VoA. When training the system hyperparameters, it has generally been observed that the training data used for the universal background model (UBM) should be coupled with that used to train the T-matrix, whereas the training data used to calculate within-class (WC) covariance matrices should be coupled to those used to generate the target models (TMs). Hereafter, we will refer to the UBM/T-matrix training and WC/TMs training stages as the front-end and back-end, respectively, as indicated in Figure 1. Accordingly, for each group of training data, three different strategies are considered:

1) *Back-to-Front*: this approach only retrains the back-end with the specified group of training data and leaves the front-end untouched (CTS-trained). This method is particularly attractive because the back-end can generally be trained relatively quickly, but requires labeled data.
2) *Front-to-Back*: this approach only retrains the front-end with the specified group of training data and leaves the back-end untouched (CTS-trained). These hyperparameters actually do not require labeled data for training so the VoA data can be used as is when it makes up a subset of the specified training data. This method is inherently included within the baseline experiments we have conducted.
3) *Hybrid training*: this approach is actually a combination of the previous two approaches. Now, all hyperparameters of the system can be retrained but with various combinations of data.

4. Results and discussion

This section presents the results for the different mismatched environment alternatives that were outlined in the previous section. In each instance, performance is assessed by means of the equal error recognition (EER) rate. In general, there are many combinations of labeling and data pooling schemes which can be considered. Here we highlight some of the main findings which demonstrate the effectiveness of the proposed approaches.

Table 1. *Baseline mismatched environment experiments.*

Standard i-vector system

	WC, TMs		
UBM, T-matrix	**CTS**	**CTS+VoA**	**VoA**
CTS	4.85%	2.53%	2.18%
CTS+VoA	5.58%	1.94%	1.82%
VoA	7.27%	1.82%	1.58%

Hybrid DNN/i-vector system

	WC, TMs		
UBM, T-matrix	**CTS**	**CTS+VoA**	**VoA**
CTS	3.39%	2.06%	2.05%
CTS+VoA	3.41%	1.55%	1.46%
VoA	6.20%	1.67%	1.62%

4.1. Baselines/domain mismatch proof-of-concept

We begin by conducting a number of experiments to establish baseline performance, assuming perfect knowledge of the VoA training data whenever needed. We considered every combination of back-end and front-end training data source as per the training groups outlined in Section 3. These baselines serve to quantify best- and worst-case performance scenarios, and as a proof-of-concept to illustrate the potential gains of incorporating the in-domain training data partition if generating accurate labels for these samples were conceivable.

The results of these baseline experiments are shown in Table 1. These results illustrate that one can greatly improve performance by incorporating the VoA data into the training, particularly when it is used in the back-end. Using the data for training the front-end alone generally degrades performance, but again improves performance when accompanied with some training on the back-end components of the system. Generally speaking, the DNN system substantially outperforms the standard i-vector system. Interestingly enough, the best performing DNN baseline is when both CTS and VoA data are used to train the front-end. One explanation for this is that the combination results in more training samples in overall, which despite some of the mismatch between domains, provides overall a richer training set for the DNN and subsequent LID system.

4.2. Score thresholding

4.2.1. Back-to-front training

Experimental results highlighting the combination of score thresholding and back-end retraining approaches are shown in Table 2. In this case, both the standard and hybrid DNN i-vector systems perform best when only VoA speech cuts corresponding to the top 75% of scores are retained for retraining. When looking at the top 50% score results, the average class purity improved but the overall system performance did not. This could be caused by the reduced sizes of some classes. Since the labels and scores are not distributed uniformly, eliminating lower scores caused certain classes to lose more cuts than others. In some cases, retraining with a combination of CTS and VoA data performed better than just training with CTS data. This is likely because the CTS data is compensating for the labeling errors in the VoA data. In this case, the best-performing DNN scenario outperforms the best-performing i-vector system scenario by 24%.

4.2.2. Hybrid training

Experimental results highlighting the combination of score thresholding and the hybrid training approach are shown in Table 3. In any case where VoA data labels are needed, the labels with the top 75% scores were used since they showed the best performance in the back-to-front experiments. The best result for both the i-vector and DNN systems occur when both CTS and VoA data are somehow combined during training, though the exact combinations differ between systems. It appears that having the labeled but mismatched CTS data somewhere in the training somehow mitigates some of the VoA labeling errors, and in general, on the order of a 25-35% improvement over the baseline mismatch scenario is achieved using this method between the different systems. Moreover, note that the DNN system outperforms the standard i-vector system by 20%.

Table 2. *Score thresholding/back-to-front results.*

Standard i-vector

Top Percentile	**VoA (EER)**	**CTS+VoA (EER)**	**Avg. VoA Class Purity**
100th	4.48%	4.30%	80.6%
75th	3.85%	4.00%	87.5%
50th	4.72%	4.24%	88.2%

Hybrid DNN/i-vector system

Top Percentile	**VoA (EER)**	**CTS+VoA (EER)**	**Avg. VoA Class Purity**
100th	3.03%	3.03%	83.0%
75th	2.91%	2.96%	86.9%
50th	2.92%	2.96%	88.4%

Table 3. *Score thresholding/hybrid training results.*

Standard i-vector system

	WC, TMs		
UBM, T-matrix	**CTS**	**CTS+VoA**	**VoA**
CTS	4.85%	4.00%	3.85%
CTS+VoA	5.58%	3.50%	3.11%
VoA	7.27%	3.76%	3.24%

Hybrid DNN/i-vector system

	WC, TMs		
UBM, T-matrix	**CTS**	**CTS+VoA**	**VoA**
CTS	3.39%	2.96%	2.91%
CTS+VoA	3.41%	2.67%	2.85%
VoA	6.20%	2.49%	2.67%

4.3. Iterative relabeling

For conciseness, we only consider DNN system performance for the remainder of this paper as it has consistently outperformed the standard i-vector system until now.

4.3.1. Back-to-front training

Experimental results highlighting the combination of iterative relabeling and the back-end retraining approach are shown in Table 4. Though this is not the case universally, we observed that the relabeling generally only improves performance if CTS data is also incorporated with VoA data to retrain the back-end. This is because the CTS data is again compensating for the labeling errors, which the relabeling mechanism may propagate from iteration to iteration. In the case where both CTS and VoA data is used, we do see an improvement in performance and improved VoA class purity at each iteration.

4.3.2. Hybrid training

Experimental results highlighting the combination of iterative relabeling and the hybrid training approach are shown in Table 5. This approach also appears to be a viable method for integrating VoA data to improve performance relative to the native CTS-only system, though its performance is slightly worse than the score thresholding/hybrid training case. The best-case performance is achieved using CTS and VoA for both the front- and back-ends, implying that the CTS data is again helping to mitigate labeling errors despite the mismatch.

4.4. Hybrid relabeling and training

We now consider the combination of both the hybrid relabeling and hybrid training techniques. In each of 3 iterations, the entire VoA training data partition is relabeled, and in any case where labeled VoA data is needed, the labels with the top 75% scores are retained. The results in Table 6 illustrate that the combination of all the techniques proposed here gives rise to the best performance of any system, yielding a 2.42% EER which is achieved when CTS and VoA are used for both the front- and back-ends. This implies that in general, both the inclusion of CTS data and score thresholding are helping to mitigate some of the labeling error which the relabeling scheme may otherwise be particularly sensitive to.

4.5. Out-of-set (OOS) experiment

Until now, we have assumed the VoA training samples are within the 9 target classes of interest. This is unlikely to be the case in practice. Lastly, we conduct an out-of-set experiment where VoA data from 9 other languages (Amharic, Creole, Croatian, English, French, Georgian, Portuguese, Turkish, and Ukrainian) is appended to the unlabeled set. We considered the most basic of the mismatch alternatives using score thresholding and back-to-front retraining, with results highlighted in Table 7. In this case, a more stringent top 50% threshold performs best as it improves label certainty in light of the OOS problem. Use of all the VoA data degrades performance relative to the 3.39% EER CTS baseline.

Table 4. *Iterative relabeling/back-to-front results.*

Experiment	EER	Avg. VoA Class Purity
Initial label	3.03%	83.0%
Relabeling (1 iteration)	2.96%	84.2%
Relabeling (3 iteration)	2.88%	85.2%
Back-to-front baseline	2.06%	100%

Table 5. *Iterative relabeling/hybrid training results.*

	WC, TMs		
UBM, T-matrix	**CTS**	**CTS+VoA**	**VoA**
CTS	3.39%	2.88%	3.06%
CTS+VoA	3.41%	2.59%	3.03%
VoA	6.20%	2.65%	2.83%

Table 6. *Hybrid labeling/hybrid training results.*

	WC, TMs		
UBM, T-matrix	**CTS**	**CTS+VoA**	**VoA**
CTS	3.39%	2.91%	3.15%
CTS+VoA	3.41%	2.42%	2.79%
VoA	6.20%	2.61%	2.82%

5. Conclusions

This paper explored alternatives for LID in mismatched environments for the case in which unlabeled in-domain data is available. The approaches varied in the ways in which data was labeled and pooled to retrain a new system. Of the two labeling schemes, score thresholding appeared to be more effective on its own. Although not the best performing scenario, improvement is gained by simply retraining the back-end only, which is useful for cases where retraining the front-end is not feasible. On the closed-set, the best performing system employed a combination of all the data labeling and pooling approaches proposed here, improving baseline out-of-domain system performance by approximately 30%. In this case, out-of-domain data must still be incorporated to counteract incurred labeling error. Experimental results also highlighted the general performance gains of the hybrid DNN/i-vector system. In the future, we intend to continue evaluating alternatives to improve the approach, particularly on the DNN side. Additionally, we plan to expand our work with the OOS case and move our experiments to harsher mismatch conditions.

Table 7. *OOS score thresholding/back-to-front results*

Top Percentile	VoA (EER)	CTS+VoA (EER)	Avg. VoA Class Purity
100th	3.65%	3.44%	48.2%
75th	3.24%	3.03%	61.5%
50th	2.97%	3.03%	78.8%

6. References

[1] N. Dehak et al., "Front-end factor analysis for speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] E. Singer et al., "The MITLL NIST LRE 2011 language recognition system," in *Proceedings of the Odyssey Workshop on Speaker and Language Recognition, June 25-28, Singapore*, 2012, pp. 209–215.

[3] P. Torres et al., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proceedings of the International Conference on Spoken Language Processing, September, Denver, CO, USA*, 2002, pp. 89–92.

[4] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.